Claims

What is claimed is:

[c1]   A method for generating information for inclusion in focus events, comprising:

maintaining a list of components requesting focus in a selected application;

determining whether a target of a first focus event matches a component at the head of the list; and

if the target of the first focus event matches the component at the head of the list, marking the component at the head of the list for inclusion in an opposite field of a second focus event.

[c2]   The method of claim 1, wherein the focus events are generated as a result of a user clicking on a focusable component.

[c3]   The method of claim 1, wherein the focus events are generated as a result of a component making a focus request through function invocation.

[c4]   The method of claim 1, wherein the target of the first focus event is the current focus owner.

[c5]   The method of claim 1, wherein determining whether the target of the first focus event matches the component at the head of the list comprises determining whether the list is empty.

[c6]   The method of claim 5, wherein marking the component at the head of the list for inclusion in the opposite field of the second focus event comprises setting the opposite field of the first focus event to null if the list is empty.

[c7]   The method of claim 5, further comprising clearing the list and setting the opposite field of the first focus event to null if the target of the first focus event does not match the component at the head of the list.

[c8] The method of claim **1**, further comprising removing the component matching the target of the first focus event from the list and marking the next component in the list as the head of the list.

[c9] The method of claim **8**, further comprising marking the component at the head of the list for inclusion in an opposite field of the first focus event.

[c10] The method of claim **9**, wherein marking the component at the head of the list for inclusion in an opposite field of the first focus event comprises determining whether the list is empty.

[c11] The method of claim **10**, wherein marking the component at the head of the list for inclusion in an opposite field of the first focus event further comprises setting the opposite field of the first focus event to null if the list is empty.

[c12] The method of claim **9**, further comprising determining whether the list is empty when a target receives the second focus event.

[c13] The method of claim **12**, further comprising setting the opposite field of the second focus event to null if the list is empty.

[c14] The method of claim **12**, further comprising determining whether the target of the second focus event matches the component at the head of the list.

[c15] The method of claim **14**, further comprising setting the opposite field of the second focus event to the component marked for inclusion in the opposite field of the second focus event if the target of the second focus event matches the component at the head of the list.

[c16] The method of claim **14**, further comprising clearing the list if the target of the second focus event does not match the component at the head of the list and setting the opposite component of the second focus event to null.

[c17] The method of claim **12**, wherein the target of the second focus event is the component gaining focus.

[c18]   The method of claim **1**, wherein maintaining the list of components comprises selectively adding a component requesting focus to the end of the list.

[c19]   The method of claim **18**, wherein selectively adding a component requesting focus to the end of the list comprises determining whether the list is empty.

[c20]   The method of claim **19**, wherein the component requesting focus is added to the end of the list if the list is empty.

[c21]   The method of claim **18**, wherein if the list is not empty, selectively adding a component requesting focus to the end of the list comprises determining whether the component requesting focus is the same as the component at the end of the list.

[c22]   The method of claim **21**, wherein the component requesting focus is added to the list if the component requesting focus is not the same as the component at the end of the list.

[c23]   A method for generating information for inclusion in focus events, the method comprising:

maintaining a list of components requesting focus in a selected application;

determining whether a target of a first focus event matches a component at the head of the list; and

if the target of the first focus event matches the component at the head of the list, marking the component at the head of the list for inclusion in an opposite field of a second focus event and marking a component next to the component at the head of the list for inclusion in an opposite field of the first focus event.

[c24]   The method of claim **23**, wherein the first focus event and the second focus event are generated as a result of a user clicking on a focusable component.

[c25]   The method of claim **23**, wherein the first focus event and the second focus event are generated as a result of a component making a focus request through function invocation.

[c26]   The method of claim **23**, wherein the target of the first focus event is the component losing focus.

[c27] The method of claim **23**, wherein determining whether the target of the first focus event matches the component at the head of the list comprises determining whether the list is empty.

[c28] The method of claim **27**, wherein marking the component next to the component at the head of the list for inclusion in the opposite field of the first focus event comprises setting the opposite field of the first focus event to null if the list is empty.

[c29] The method of claim **27**, further comprising clearing the list and setting the opposite field of the first focus event to null if the target of the first focus event does not match the component at the head of the list.

[c30] The method of claim **23**, wherein marking the next component for inclusion in the opposite field of the first focus event comprises removing the component matching the target of the first focus event from the list and subsequently determining whether the list is empty.

[c31] The method of claim **30**, wherein marking the next component for inclusion in the opposite field of the first focus event further comprises setting the opposite field of the first focus event to null if the list is empty.

[c32] The method of claim **23**, further comprising determining whether the list is empty when a target receives the second focus event.

[c33] The method of claim **32**, further comprising setting the opposite field of the second focus event to null if the list is empty.

[c34] The method of claim **32**, further comprising determining whether the target of the second focus event matches the component at the head of the list.

[c35] The method of claim **34**, further comprising setting the opposite field of the second focus event to the component marked for inclusion in the opposite field of the second focus event if the target of the second focus event matches the component at the head of the list.

17

[c36] The method of claim **34**, further comprising clearing the list if the target of the second focus event does not match the component at the head of the list and setting the opposite component of the second focus event to null.

[c37] The method of claim **32**, wherein the target of the second focus event is the component gaining focus.

[c38] The method of claim **23**, wherein maintaining the list of components comprises selectively adding a component requesting focus to the end of the list.

[c39] A computer-readable medium having stored thereon a program which is executable by a processor, the program comprising instructions for:

maintaining a list of components requesting focus in a selected application; and

determining an opposite field of a first focus event and an opposite of a second focus event based on a target of the first focus event, a target of the second focus event, and the list of components requesting focus.

[c40] The computer-readable medium of claim **39**, wherein the focus events are generated as a result of operation of the selected application.

[c41] The computer-readable medium of claim **40**, wherein the focus events are generated as a result of a user selecting a focusable component in the selected application.

[c42] The computer-readable medium of claim **40**, wherein the focus events are generated as a result of a component in the selected application making a focus request through function invocation.

[c43] The computer-readable medium of claim **39**, wherein the program marks a component at the head of the list for inclusion in the opposite field of the second focus event if the target of the first focus event matches the component at the head of the list.

[c44] The computer-readable medium of claim **43**, wherein the program marks a component next to the component at the head of the list for inclusion in the opposite field of the first focus event.

[c45] The computer-readable medium of claim **39**, wherein maintaining the list of components comprises selectively adding a component requesting focus to the end of the list and selectively removing a component matching a target of the first focus event from the head of the list.

[c46] The computer-readable medium of claim **39**, wherein the target of the first focus event is the current focus owner and the target of the second focus event is the component gaining focus.